IN THE CLAIMS:

This listing of claims will replace all prior versions, and listing, of claims in the application:

1. (previously presented) A method of managing a network switch having a processor card including a memory and a processing unit in the processor card comprising:

detecting an error;

determining if the error is ignorable;

determining whether a threshold has been reached if the error is determined to be ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

performing a hitless rebuild in the processor card if the threshold has not been reached.

2. (previously presented) The method of claim 1, wherein the performing of the hitless rebuild includes:

performing an initialization of the memory; and,

protecting a portion of the memory from access by the processing unit during the initialization.

3. (previously presented) The method of claim 1, wherein the performing of the includes protecting a portion of the memory that contains a set of routing tables.

4. (previously presented) The method of claim 1, wherein the performing of the hitless rebuild further comprises protecting a portion of the memory that contains a set of state tables.

5. (previously presented) The method of claim 1, wherein the memory is accessed through a set of memory addresses and the performing of the hitless rebuild further comprises preventing the processing unit from accessing a predetermined set of memory addresses in the set of memory addresses.

6. (canceled)

7. (previously presented) The method of claim 1, further comprising setting the processing unit to enter into a degraded mode if the error is not ignorable and if the threshold has been reached.

8. (canceled)

9. (presently presented) An apparatus for managing a network switch having a processor card including a memory and a processing unit in the processor card comprising:

means for detecting an error;

means for determining if the error is ignorable;

means for determining whether a threshold has been reached if the error is determined to be ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

means for performing a hitless rebuild in the processor card if the threshold has not been reached.

10. (previously presented) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

detecting an error;

determining if the error is an ignorable error;

determining whether a threshold has been reached if the error is determined to be an ignorable, the threshold corresponding to a number of hitless rebuilds that have occurred within an amount of time; and,

performing a hitless rebuild in a processor card if the threshold has not been reached.

11. (previously presented) The article of claim 10, wherein the method further comprises:

performing an initialization of the memory; and,

protecting a portion of the memory from access by the processing unit during the initialization.

12. (previously presented) The article of claim 10, wherein the method further comprises:

protecting a portion of the memory that contains a set of routing tables.

13. (previously presented) The article of claim 10, wherein the method further comprises:

protecting a portion of the memory that contains a set of state tables.

14. (previously presented) The article of claim 10, wherein thememory further comprises:

preventing a processing unit from accessing a predetermined set of memory addresses in the set of memory addresses.

15. (canceled)

16. (previously presented) The article of claim 10, wherein the method further comprises:

setting of the processing unit to enter into a degraded mode if the threshold has been reached.:

17. (canceled)

18. (currently amended) A method, comprising:

re-initializing software that is executed on a ~~networking device~~ card located within a networking hardware apparatus, said re-initializing in response to an error, said ~~networking device~~ card comprising a processor that executes said software, said ~~networking device~~ card comprising a protected memory region that stores information that can be used by said processor to execute said software after said software has been re-initialized, said information further

comprising a routing table information, said re-initializing not deleting said routing table information from said protected memory region.

19. (currently amended) The method of claim 18 wherein said protected memory region further comprises non-volatile memory.

20. (currently amended) The method of claim 18 wherein said protected memory region further comprises volatile memory.

21. (currently amended) The method of claim 20 wherein said ~~volatile memory~~ protected memory region further comprises a segment of random access memory, said not deleting further comprising said processor not accessing said segment during said re-initializing for purposes of clearing said segment during said re-initializing.

22. (currently amended) The method of claim ~~21~~ 20 wherein said protected memory region further comprises a segment of random access memory and said card comprises a memory management unit to manage said protected memory region, said not deleting further comprising said memory management unit not allowing access to said segment so as to prevent said segment from being cleared during said re-initializing ~~random access memory further comprises dynamic random access memory~~.

23. (currently amended) A method, comprising:

re-initializing software that is executed on a ~~networking device~~ card located within a networking hardware apparatus, said re-initializing in response to an error, said ~~networking device~~ card comprising a processor that executes said software, said ~~networking device~~ card comprising a protected memory region that stores information that can be used by said processor to execute said software after said software has been re-initialized, said information further comprising ~~a~~ state table information, said state table information further comprising the status of an interface card located within said networking hardware apparatus, said re-initializing not deleting said ~~state table information~~ status from said protected memory region.

24. (currently amended) The method of claim 23 wherein said protected memory region further comprises non-volatile memory.

25. (currently amended) The method of claim 23 wherein said protected memory region further comprises volatile memory.

26. (currently amended) The method of claim 25 wherein said ~~volatile memory~~ protected memory region further comprises a segment of random access memory, said not deleting further comprising said processor not accessing said segment during said re-initializing for purposes of clearing said segment during said re-initializing.

27. (currently amended) The method of claim ~~26~~ 25 wherein said protected memory region further comprises a segment of random access memory and said card comprises a memory management unit to manage said protected memory

region, said not deleting further comprising said memory management unit not allowing access to said segment so as to prevent said segment from being cleared during said re-initializing ~~random access memory further comprises dynamic random access memory~~.

28. (currently amended) An apparatus, comprising:

a ~~networking device~~ card for use in a networking hardware apparatus, said ~~networking device~~ card comprising a processor ~~that~~ to executes re-initializable software, said ~~networking device~~ card comprising ~~a~~ memory, said memory comprising a protected memory region to ~~that stores~~ information that can be used by said processor to execute said re-initializable software after said software has been re-initialized, said information further comprising ~~a~~ routing table information, said re-initializable software ~~able to be~~ comprising instructions to for re-initializing~~d~~ said software in response to an error, said instructions executable by said processor, ~~in a manner that does not cause~~ said routing table information ~~to be~~ not being deleted from said protected memory region during said re-initializing.

29. (currently amended) The ~~method~~ apparatus of claim 28 wherein said protected memory region further comprises non-volatile memory.

30. (currently amended) The ~~method~~ apparatus of claim 28 wherein said protected memory region further comprises volatile memory.

31. (currently amended) The ~~method~~ apparatus of claim 30 wherein said ~~volatile memory~~ protected memory region further comprises a segment of random access memory, said processor not accessible to said segment during said re-initializing so as to prevent said protected memory region from being cleared during said re-initializing.

32. (currently amended) The ~~method~~ apparatus of claim ~~31~~ 30 wherein said protected memory region further comprises a segment of random access memory and said card further comprises a memory management unit, said memory management unit to not allow said segment to be accessed during said re-initializing for purposes of clearing said protected memory region ~~random access memory further comprises dynamic random access memory~~.

Cont 1

33. (currently amended) An apparatus, comprising:

a ~~networking device~~ card for use in a networking hardware apparatus, said ~~networking device~~ card comprising a processor ~~that~~ to executes re-initializable software, said ~~networking device~~ card comprising a memory, said memory comprising a protected memory region ~~that~~ to stores information that can be used by said processor to execute said re-intializable software after said software has been re-initialized, said information further comprising a state table information, said state table information further comprising the status of an interface card located within said networking hardware apparatus, said re-initializable software comprising instructions ~~able to be re-initialized~~ for re-

initializing said software in response to an error, ~~in a manner that does not cause~~ said ~~state table information to be~~ status not being deleted from said protected memory region during said re-initializing.

34. (currently amended) The ~~method~~ apparatus of claim 33 wherein said protected memory region further comprises non-volatile memory.

35. (currently amended) The ~~method~~ apparatus of claim 33 wherein said protected memory region further comprises volatile memory.

Cont D'

36. (currently amended) The ~~method~~ apparatus of claim 35 wherein said ~~volatile memory~~ protected memory region further comprises a segment of random access memory, said processor not accessible to said segment during said re-initializing so as to prevent said protected memory region from being cleared during said re-initializing.

37. (currently amended) The ~~method~~ apparatus of claim ~~36~~ 35 wherein said protected memory region further comprises a segment of random access memory and said card further comprises a memory management unit, said memory management unit to not allow said segment to be accessed during said re-initializing for purposes of clearing said protected memory region ~~random access memory further comprises dynamic random access memory~~.

38. (currently amended) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

re-initializing software that is executed on a ~~networking device~~ card within a networking hardware apparatus, said re-initializing in response to an error, said ~~networking device~~ card comprising a processor that executes said software, said ~~networking device~~ card comprising a memory that stores information that can be used by said processor to execute said software after said software has been re-initialized, said information further comprising ~~a~~ routing table information, said re-initializing not deleting said routing table information from said memory because said instructions are written so as to not access a segment of said memory during said re-initializing.

Cont D'

39. (previously added) The article of claim 38 wherein said memory further comprises non-volatile memory.

40. (previously added) The article of claim 38 wherein said article is non-volatile memory.

41. (previously added) The article of claim 38 wherein said memory further comprises volatile memory.

42. (previously added) The article of claim 41 wherein said volatile memory further comprises random access memory.

43. (previously added) The article of claim 42 wherein said random access memory further comprises dynamic random access memory.

44. (currently amended) An article comprising a machine readable medium having instructions stored thereon which, when executed, cause a method to be performed, the method comprising:

re-initializing software that is executed on a networking device card, said re-initializing in response to an error, said networking device card comprising a processor that executes said software, said networking device card comprising a memory that stores information that can be used by said processor to execute said software, said information further comprising ~~a~~ state table information, said re-initializing not deleting said state table information from said memory because said instructions are written so as to not access a segment of said memory during said re-initializing.

45. (previously presented) The article of claim 44 wherein said memory further comprises non-volatile memory.

46. (previously presented) The article of claim 44 wherein said article is non-volatile memory.

47. (previously presented) The article of claim 44 wherein said memory further comprises volatile memory.

48. (previously presented) The article of claim 47 wherein said volatile memory further comprises random access memory.

49. (previously presented) The article of claim 48 wherein said random access memory further comprises dynamic random access memory.

50. (currently amended) An apparatus, comprising:

means for re-initializing software that is executed on a ~~networking device~~ card located within a networking hardware apparatus, said re-initializing in response to an error, said ~~networking device~~ card comprising a processor ~~that~~ to executes said software, said networking device card comprising a protected memory region to ~~that~~ stores information that can be used by said processor to execute said software after said software has been re-initialized, said information further comprising a routing table information, said means for re-initializing further comprising means for not deleting said routing table information from said protected memory region during said re-initializing.

Cont D'

51. (previously presented) The apparatus of claim 50 wherein said memory further comprises non-volatile memory.

52. (previously presented) The apparatus of claim 50 wherein said memory further comprises volatile memory.

53. (previously presented) The apparatus of claim 52 wherein said volatile memory further comprises random access memory.

54. (previously presented) The apparatus of claim 53 wherein said random access memory further comprises dynamic random access memory.

55. (currently amended) An apparatus, comprising:

means for re-initializing software that is executed on a ~~networking device~~ card located within a networking hardware apparatus, said re-initializing in response to an error, said ~~networking device~~ card comprising a processor ~~that~~ to executes said software, said networking device card comprising a protected memory ~~that~~ region to stores information that can be used by said processor to execute said software, said information further comprising a state table information, said state table information further comprising the status of an interface card located within said networking hardware apparatus, said means for re-initializing further comprising means for not deleting said ~~state table information~~ status from said protected memory region during said re-initializing.

56. (previously presented) The apparatus of claim 55 wherein said memory further comprises non-volatile memory.

57. (previously presented) The apparatus of claim 55 wherein said memory further comprises volatile memory.

58. (previously presented) The apparatus of claim 57 wherein said volatile memory further comprises random access memory.

59. (previously presented) The apparatus of claim 58 wherein said random access memory further comprises dynamic random access memory.

60. (new) A method, comprising:

re-initializing software that is executed on a card located within a networking hardware apparatus, said re-initializing in response to an error, said card comprising a processor that executes said software, said card comprising a protected memory region that stores information that can be used by said processor to execute said software after said software has been re-initialized, said information further comprising state table information, said state table information further comprising network topology information, said re-initializing not deleting said network topology information from said protected memory region.

61. (new) The method of claim 60 wherein said protected memory region further comprises non-volatile memory.

62. (new) The method of claim 60 wherein said protected memory region further comprises volatile memory.

63. (new) The method of claim 62 wherein said protected memory region further comprises a segment of random access memory, said not deleting further comprising said processor not accessing said segment during said re-initializing for purposes of clearing said segment during said re-initializing.

64. (new) The method of claim 62 wherein said protected memory region further comprises a segment of random access memory and said card comprises a memory management unit to manage said protected memory region, said not deleting further comprising said memory management unit not allowing access to said segment so as to prevent said segment from being cleared during said re-initializing.

Cont D'

65. (new) An apparatus, comprising:

a card for use in a networking hardware apparatus, said card comprising a processor to execute re-initializable software, said card comprising memory, said memory comprising a protected memory region to store information that can be used by said processor to execute said re-intializable software after said software has been re-initialized, said information further comprising state table information, said state table information further comprising network topology information, said re-initializable software comprising instructions for re-initializing said software in response to an error, said network topology information not being deleted from said protected memory region during said re-initializing.

66. (new) The apparatus of claim 65 wherein said protected memory region further comprises non-volatile memory.

67. (new) The apparatus of claim 65 wherein said protected memory region further comprises volatile memory.

68. (new) The apparatus of claim 67 wherein said protected memory region further comprises a segment of random access memory, said processor not accessible to said segment during said re-initializing so as to prevent said protected memory region from being cleared during said re-initializing.

69. (new) The apparatus of claim 67 wherein said protected memory region further comprises a segment of random access memory and said card further comprises a memory management unit, said memory management unit to not allow said segment to be accessed during said re-initializing for purposes of clearing said protected memory region.